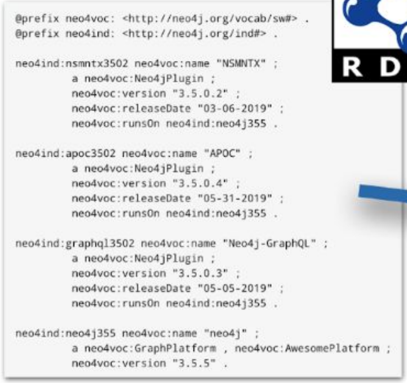
Going Meta #12: Importing RDF data into Aura with Python + RDFLib

What we know… (n10s)

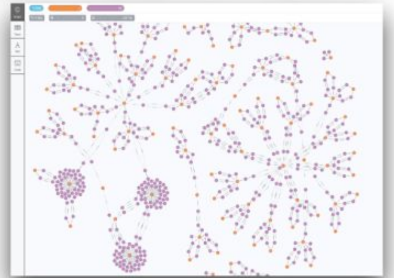


```
@prefix neo4voc: <http://neo4j.org/vocab/sw#> .
@prefix neo4ind: <http://neo4j.org/ind#> .

neo4ind:nsmntx3502 neo4voc:name "NSMNTX" ;
          a neo4voc:Neo4jPlugin ;
          neo4voc:version "3.5.0.2" ;
          neo4voc:releaseDate "03-06-2019" ;
          neo4voc:runsOn neo4ind:neo4j355 .

neo4ind:apoc3502 neo4voc:name "APOC" ;
          a neo4voc:Neo4jPlugin ;
          neo4voc:version "3.5.0.4" ;
          neo4voc:releaseDate "05-31-2019" ;
          neo4voc:runsOn neo4ind:neo4j355 .

neo4ind:graphql3502 neo4voc:name "Neo4j-GraphQL" ;
          a neo4voc:Neo4jPlugin ;
          neo4voc:version "3.5.0.3" ;
          neo4voc:releaseDate "05-05-2019" ;
          neo4voc:runsOn neo4ind:neo4j355 .

neo4ind:neo4j355 neo4voc:name "neo4j" ;
          a neo4voc:GraphPlatform , neo4voc:AwesomePlatform ;
          neo4voc:version "3.5.5" .
```

neosemantics

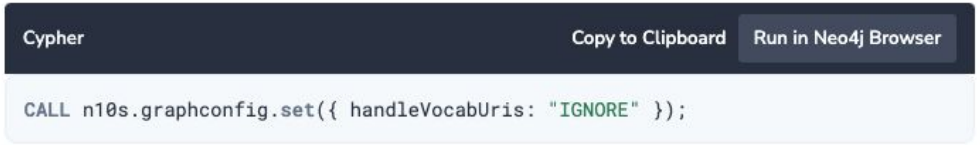
```
CALL n10s.rdf.import.fetch("https://.../nsmntx.ttl","Turtle")
```

neo4j

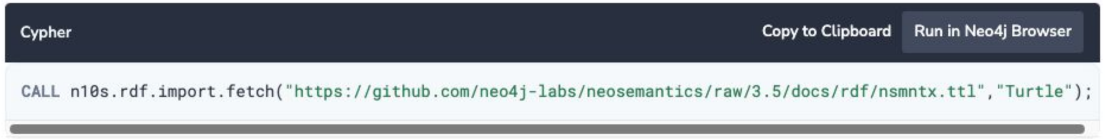


```
MATCH (n:Neo4jPlugin)-[:runsOn]->(p:GraphPlatform)
WHERE p.name = "neo4j"
RETURN n.name, MAX(n.version)
```

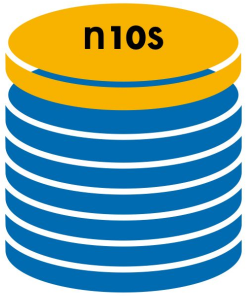
The “limitation”



```
CALL n10s.graphconfig.set({ handleVocabUris: "IGNORE" });
```



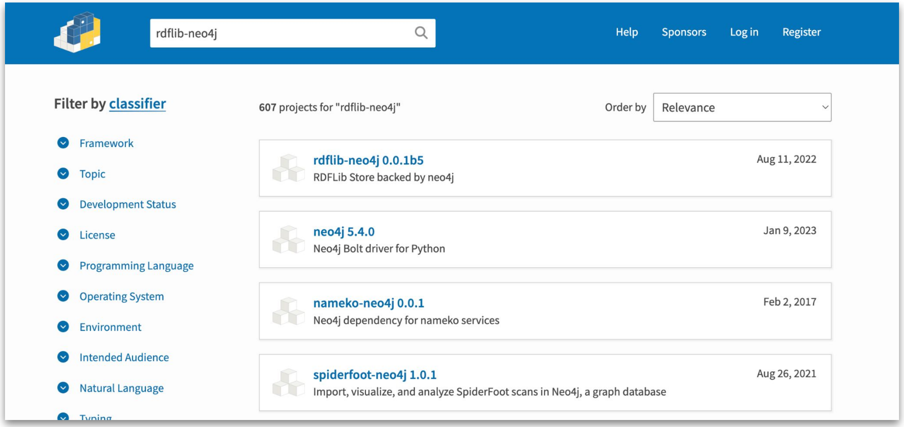
```
CALL n10s.rdf.import.fetch("https://github.com/neo4j-labs/neosemantics/raw/3.5/docs/rdf/nsmntx.ttl","Turtle");
```



n10s deployed on the server side

The solution: rdflib-neo4j package

`https://pypi.org/search/?q=rdflib-neo4j`



rdflib-neo4j

Help | Sponsors | Log in | Register

Filter by classifier

- Framework
- Topic
- Development Status
- License
- Programming Language
- Operating System
- Environment
- Intended Audience
- Natural Language
- Typing

607 projects for "rdflib-neo4j"

Order by: Relevance

rdflib-neo4j 0.0.1b5 — Aug 11, 2022
RDFLib Store backed by neo4j

neo4j 5.4.0 — Jan 9, 2023
Neo4j Bolt driver for Python

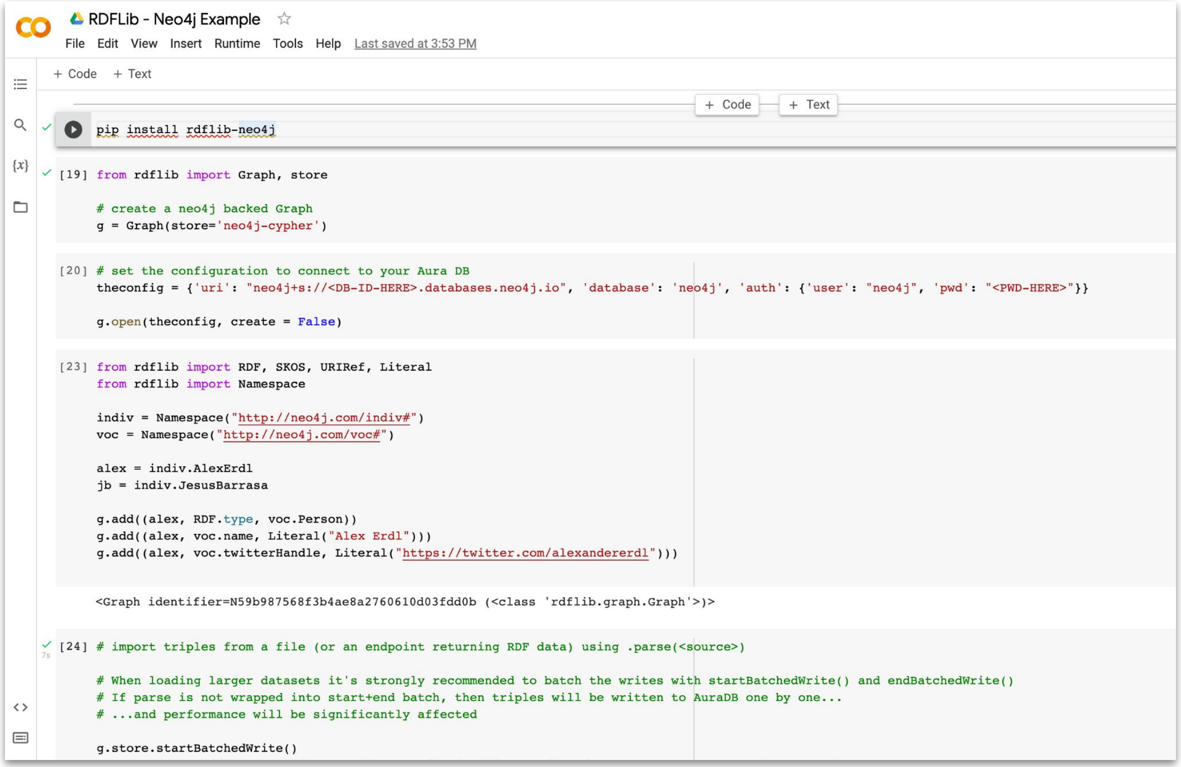
nameko-neo4j 0.0.1 — Feb 2, 2017
Neo4j dependency for nameko services

spiderfoot-neo4j 1.0.1 — Aug 26, 2021
Import, visualize, and analyze SpiderFoot scans in Neo4j, a graph database

```
pip install rdflib-neo4j
```

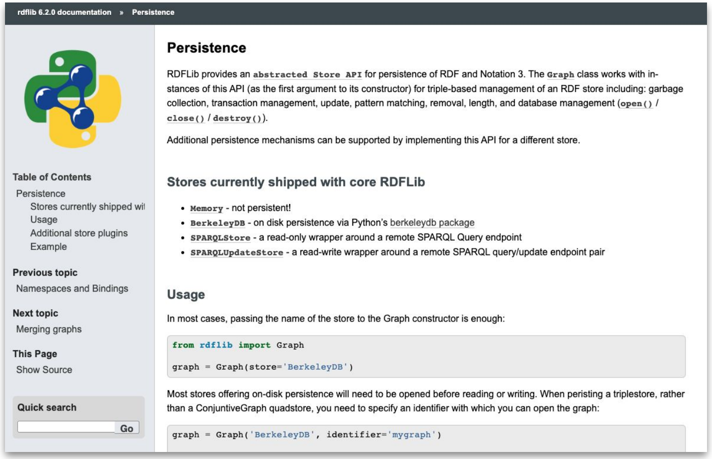
```python
from rdflib import Graph, store

# create a neo4j backed Graph
g = Graph(store='neo4j-cypher')
```

Let’s see it in action


RDFLib - Neo4j Example
File Edit View Insert Runtime Tools Help Last saved at 3:53 PM
+ Code + Text
+ Code + Text

```
pip install rdflib-neo4j

[19] from rdflib import Graph, store

     # create a neo4j backed Graph
     g = Graph(store='neo4j-cypher')

[20] # set the configuration to connect to your Aura DB
     theconfig = {'uri': "neo4j+s://<DB-ID-HERE>.databases.neo4j.io", 'database': 'neo4j', 'auth': {'user': "neo4j", 'pwd': "<PWD-HERE>"}}

     g.open(theconfig, create = False)

[23] from rdflib import RDF, SKOS, URIRef, Literal
     from rdflib import Namespace

     indiv = Namespace("http://neo4j.com/indiv#")
     voc = Namespace("http://neo4j.com/voc#")

     alex = indiv.AlexErdl
     jb = indiv.JesusBarrasa

     g.add((alex, RDF.type, voc.Person))
     g.add((alex, voc.name, Literal("Alex Erdl")))
     g.add((alex, voc.twitterHandle, Literal("https://twitter.com/alexandererdl")))

     <Graph identifier=N59b987568f3b4ae8a2760610d03fdd0b (<class 'rdflib.graph.Graph'>)>

[24] # import triples from a file (or an endpoint returning RDF data) using .parse(<source>)

     # When loading larger datasets it's strongly recommended to batch the writes with startBatchedWrite() and endBatchedWrite()
     # If parse is not wrapped into start+end batch, then triples will be written to AuraDB one by one...
     # ...and performance will be significantly affected

     g.store.startBatchedWrite()
```

Under the hood



rdflib 6.2.0 documentation » Persistence

Table of Contents
- Persistence
  - Stores currently shipped wi
  - Usage
  - Additional store plugins
  - Example

Previous topic
Namespaces and Bindings

Next topic
Merging graphs

This Page
Show Source

Quick search [Go]

Persistence

RDFLib provides an `abstracted Store API` for persistence of RDF and Notation 3. The `Graph` class works with instances of this API (as the first argument to its constructor) for triple-based management of an RDF store including: garbage collection, transaction management, update, pattern matching, removal, length, and database management (`open()` / `close()` / `destroy()`).

Additional persistence mechanisms can be supported by implementing this API for a different store.

Stores currently shipped with core RDFLib

- `Memory` - not persistent!
- `BerkeleyDB` - on disk persistence via Python's berkeleydb package
- `SPARQLStore` - a read-only wrapper around a remote SPARQL Query endpoint
- `SPARQLUpdateStore` - a read-write wrapper around a remote SPARQL query/update endpoint pair

Usage

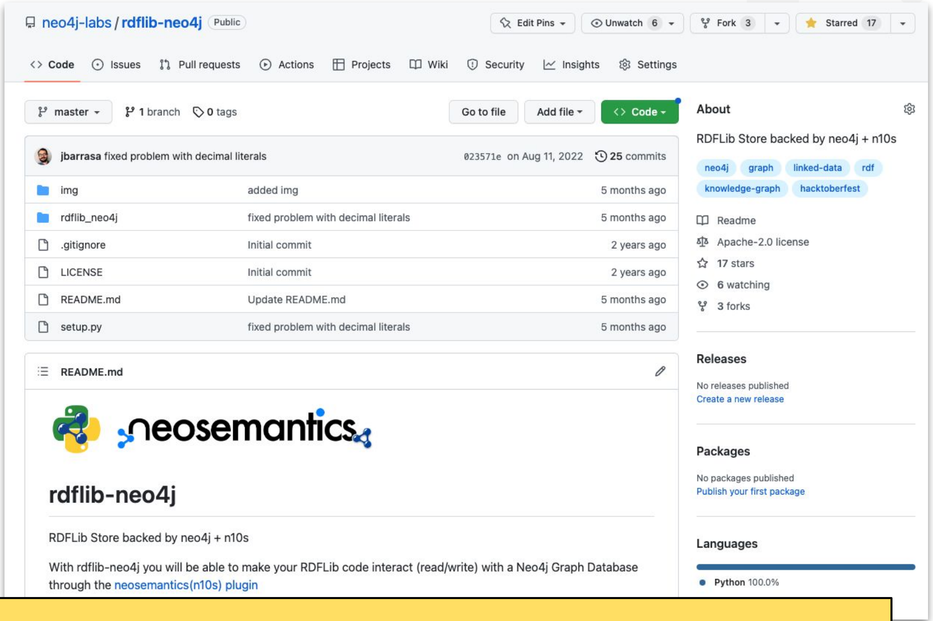
In most cases, passing the name of the store to the Graph constructor is enough:

```
from rdflib import Graph

graph = Graph(store='BerkeleyDB')
```

Most stores offering on-disk persistence will need to be opened before reading or writing. When peristing a triplestore, rather than a ConjuntiveGraph quadstore, you need to specify an identifier with which you can open the graph:

```
graph = Graph('BerkeleyDB', identifier='mygraph')
```

`https://rdflib.readthedocs.io/en/stable/persistence.html`



neo4j-labs / rdflib-neo4j Public

Edit Pins · Unwatch 6 · Fork 3 · Starred 17

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

master · 1 branch · 0 tags · Go to file · Add file · Code

jbarrasa fixed problem with decimal literals · 023571e on Aug 11, 2022 · 25 commits

img	added img	5 months ago
rdflib_neo4j	fixed problem with decimal literals	5 months ago
.gitignore	Initial commit	2 years ago
LICENSE	Initial commit	2 years ago
README.md	Update README.md	5 months ago
setup.py	fixed problem with decimal literals	5 months ago

README.md

neosemantics

rdflib-neo4j

RDFLib Store backed by neo4j + n10s

With rdflib-neo4j you will be able to make your RDFLib code interact (read/write) with a Neo4j Graph Database through the neosemantics(n10s) plugin

About

RDFLib Store backed by neo4j + n10s

neo4j · graph · linked-data · rdf · knowledge-graph · hacktoberfest

Readme
Apache-2.0 license
17 stars
6 watching
3 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Python 100.0%

Shall we have a look at the source?